1983

Modeling Multimicrocomputer Networks

Daniel A. Reed

Report Number:
83-436

Modeling Multimicrocomputer Networks

Daniel A. Reed

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD-TR-436

ABSTRACT

Recent developments in very large scale integration have made it feasible to construct a highly parallel computer composed of large numbers of interconnected microcomputers. The individual nodes of these multimicrocomputer networks do not share any common memory, making it crucial to select an interconnection network capable of efficiently supporting internode communication. The modeling problems posed by this approach to parallel processing differ in several significant respects from those associated with traditional queueing network models of computer systems. Among them are the size of the models, the varied interconnection network topologies, and algorithm dependent internode communication patterns. We extend queueing theoretic models to include multimicrocomputer networks with primary emphasis on two areas: characterizing interconnection network workloads and finding efficient solution techniques for the resulting models.

March 8, 1983

Introduction

Dramatic advances in very large scale integration have suggested a new paradigm for parallel computation based on large numbers of interconnected microcomputer nodes. Each network node, fabricated as one or two VLSI chips, would contain a processor with some locally addressable memory, a communication controller capable of routing messages without delaying the processor, and a small number of connections to other nodes.

Proposed application areas for multimicrocomputer networks have included mini-max game tree searches [ALK80], finite element problems [SMIT82], and partial differential equations solvers [REED83b]. The cooperating tasks of parallel algorithms for these problems would execute asynchronously on different nodes and communicate via message passing. Because the nodes do not share any globally accessible memory, the performance of a multimicrocomputer network is critically dependent on the underlying interconnection network and its message passing efficiency. Furthermore, the VLSI implementation of the nodes and their limited fanout severely constrain the interconnection network connectivity.

The modeling problems posed by this approach to parallel processing differ in several significant respects from those associated with traditional queueing network models of computer systems [BUZE71]. Among the important differences are:

- model size,
- varied interconnection network topologies, and

- algorithm dependent communication patterns.

By most standards, a computer system model containing one hundred devices would be considered extremely large, but multimicrocomputer networks containing up to 10^6 nodes have been proposed [DESP78, WITT81], with a one hundred node network classified as small. With the inclusion of an interconnection network in the model, techniques capable of accommodating vast numbers of devices are needed.

The performance of an algorithm executing on a multimicrocomputer network depends not only on the internode communication pattern of the algorithm but also on how well the topology of the network (e.g., ring or nearest neighbor) matches the communication pattern. Useful models should include the influence of both communication patterns and network topology on system performance.

In this paper we extend queueing theoretic models to include multimicrocomputer networks. Primary attention is focused on two areas: characterizing interconnection network workloads and finding efficient solution techniques for multimicrocomputer network models. We illustrate the techniques using one possible network: the bidirectional ring. As we shall see, the ring is generally not a desirable interconnection strategy for large multimicrocomputer networks; we use it simply for pedagogic purposes.

Network Communication Patterns

Each message generated by a source node crosses some communication links and passes through the communication controllers of intermediate nodes before reaching its destination node and causing some computation to occur. Each communication link crossing and the resulting computation constitute a

visit to a link or node.

Given a multimicrocomputer network with a specified interconnection network topology and the internode communication pattern of an algorithm, it is in principle possible to calculate the mean number of visits made by a message to each device. In practice however, the size of the network generally makes this calculation both computationally intractable and numerically unstable; we must elide some details and adopt an abstraction of the actual system. Because the selection of an appropriate interconnection network is so crucial, we have chosen to abstract the communication patterns while retaining representation of interconnection network details.

In its most general form, the mean number of link visits, LV, made by a message is given by

$$LV = \sum_{l=1}^{lmax} l \cdot \Phi(l) \ , \tag{1}$$

where $\Phi(l)$ is the probability of a message crossing l communication links, and $lmax$ is the maximum internode distance. Different choices for $\Phi(l)$ lead to different message routing distributions and in turn, different values of LV. In the following, we consider three different message routing distributions for which it is possible to obtain closed forms for LV. To do this, however, we must first distinguish between two types of interconnection networks: symmetric and asymmetric.

In a *symmetric* interconnection network there exists a homomorphism that maps any node of the network graph onto any other node. Intuitively, all nodes possess the same view of the network. A bidirectional ring is a simple example of a symmetric interconnection network because each message can always

reach two nodes by crossing any given number of communication links. Conversely, an *asymmetric* interconnection network is any network that is *not* symmetric; a tree structured interconnection is an example. For simplicity's sake, our discussion is limited to symmetric networks; see [REED83a] for a discussion of asymmetric networks. Table I summarizes the notation using in describing communication patterns for symmetric networks.

Uniform Message Routing

Message routing is said to be *uniform* if the probability of node i sending a message to node j is the same for all i and j, $i \neq j$. Because we are interested in message transfers requiring use of the interconnection network, we exclude the case in which nodes send messages to themselves.

Consider an interconnection network containing K nodes obeying uniform message routing. Define $Reach(l, Net-type)$ as the number of nodes reachable from an arbitrary node by crossing exactly l communication links. Then the probability of a message requiring l link traversals to reach its destination is

$$\Phi(l) = \frac{Reach(l, Net-type)}{K - 1},$$

and the mean number of links traversed by a message is

$$L_{V}^{uniform} = \frac{\sum_{l=1}^{lmax} l \cdot Reach(l, Net-type)}{K - 1},$$

where $lmax$ is the maximum distance to any node.

The uniform message routing distribution is appealing because it makes no assumptions about the nature of the computation generating the messages. Since most distributed computations should exhibit some measure of locality in

the exchange of messages, it also provides what is likely to be an upper bound on the mean number of link visits.

As an illustration, consider the bidirectional ring with an odd number of nodes K. For any specified distance in each direction from a source node, two other nodes can be reached. Thus,

$$Reach(l, ring) = \begin{cases} 2 & 1 \leq l \leq \left\lfloor \frac{K}{2} \right\rfloor \\ 0 & l > \left\lfloor \frac{K}{2} \right\rfloor . \end{cases}$$

and

$$LV_{ring}^{uniform} = \frac{\sum_{l=1}^{\left\lfloor \frac{K}{2} \right\rfloor} 2l}{K - 1} = \frac{K + 1}{4} .$$

Sphere of Locality Message Routing

Suppose the uniform message routing assumption were relaxed. Any reasonable mapping of a distributed computation onto a multimicrocomputer network should place those tasks that exchange messages with high frequency physically close to one another in the network. One abstraction of this idea places each node at the center of a sphere of locality. A node sends messages to the other nodes in its sphere of locality with some high probability φ, whereas messages are sent to nodes outside the sphere of locality with low probability $1 - \varphi$.

If L is the maximum number of links a message may cross and remain in the sphere of locality centered at its source (i.e., L is the radius of the sphere), the number of nodes contained in a sphere of locality is

$$LocSize(L, Net-type) = \sum_{l=1}^{L} Reach(l, Net-type) .$$

The network symmetry implies that each node is contained in the localities of $LocSize(L, Net-type)$ other nodes and is outside the localities of $K - LocSize(L, Net-type) - 1$ nodes. Thus, the message traffic on the communication links is uniform even though the message routing distribution is not.

Given the values of φ and L, the message routing distribution is given by

$$\Phi(l) = \begin{cases} \dfrac{\varphi\ Reach(l, Net-type)}{LocSize(L, Net-type)} & 1 \le l \le L \\ \dfrac{(1-\varphi)\ Reach(l, Net-type)}{K - LocSize(L, Net-type) - 1} & L < l \le lmax \end{cases} ,$$

and the mean number of communication links traversed by a message under this message routing distribution is

$$LV^{local} = \sum_{l=1}^{lmax} l \cdot \Phi(l) \qquad (2)$$

$$= \frac{\varphi \sum_{l=1}^{L} l \cdot Reach(l, Net-type)}{LocSize(L, Net-type)}$$

$$+ \frac{(1-\varphi)\left[LV^{uniform}\ (K-1) - \sum_{l=1}^{L} l \cdot Reach(l, Net-type)\right]}{K - LocSize(L, Net-type) - 1} .$$

The first term is simply the product of the average number of links traversed

when sending a message to a node in the locality and the probability of visiting the locality φ. The second term has a similar interpretation for nodes outside the locality.

For the bidirectional ring, the locality size is merely $2L$, and

$$\Phi(l) = \begin{cases} \dfrac{\varphi}{L} & 1 \le l \le L \\ \dfrac{2(1-\varphi)}{K-2L-1} & L < l \le \left\lfloor \dfrac{K}{2} \right\rfloor . \end{cases}$$

Substituting this function into (2) gives

$$LV_{ring}^{local} = \frac{\varphi(L+1)}{2} + \frac{(1-\varphi)\left[(K-1)^2 - 4L(L+1)\right]}{4K(K-2L-1)} .$$

Decreasing Probability Message Routing

The previous definition of locality is useful if the probability of visiting the locality is high and the size of the locality is small compared to the size of the network. There are, however, many cases when this is not an appropriate abstraction. An alternative, intuitively appealing, notion of locality is that the probability of sending a message to a node decreases as the distance of the destination node from the source node increases.

A wide variety of distribution functions exhibiting some rate of decay exist, but the distribution function

$$\Phi(l) = Decay(d, lmax) \cdot d^l \qquad 0 < d < 1 ,$$

where $Decay(d, lmax)$ is a normalizing constant, is particularly attractive. As d

approaches one, the cdf of Φ approximates a linearly increasing function of the distance from the source node. Conversely, as d approaches zero, the cdf of Φ approaches a nearest neighbor communication pattern. Choices of d between these two extremes lead to varying degrees of message routing locality.

The value of $Decay(d, lmax)$ is chosen so that

$$Decay(d, lmax) \sum_{l=1}^{lmax} d^l = 1 .$$

Simple algebra yields

$$Decay(d, lmax) = \frac{d - 1}{d(d^{lmax} - 1)} ,$$

and

$$\Phi(l) = \frac{(d - 1)d^l}{d^{lmax} - 1} .$$

Substituting this into (1), we obtain

$$LV^{decay} = \sum_{l=1}^{lmax} l \cdot \Phi(l) \qquad (3)$$

$$= \frac{(d - 1)}{(d^{lmax} - 1)} \sum_{l=1}^{lmax} l \cdot d^l$$

$$= \frac{d\left[(d \cdot lmax - lmax - 1)d^{lmax} + 1\right]}{(d - 1)(d^{lmax} - 1)} .$$

Interestingly, $lmax$ is the only network dependent parameter in this formula.

Finally, LV for the bidirectional ring is obtained by substituting $\left\lfloor \frac{K}{2} \right\rfloor$ for $lmax$ in (3).

Figure I shows the mean number of link visits for a fifteen node ring under the various message routing distributions. By varying the parameters of these distributions, a wide variety of distribution functions ranging from uniform message routing to nearest neighbor communication can be obtained.

Visit Ratios and Service Times

In our formulation of message routing in the previous section, a message leaves a source node, crosses some communication links to reach its destination, and causes some computation to take place there. The visit ratio V_i is defined as the average number of visits made to device i by a message.

Under the uniform message routing distribution discussed earlier, it is obvious that the visit ratios for all network nodes must be the same. Because a message visits only one node, its destination, we require the sum of the node visit ratios to be one. This immediately leads to the following value for the node visit ratios:

$$V_{PE} = \frac{1}{K} . \tag{4}$$

Somewhat surprisingly, the node visit ratios are also given by (4) when the two non-uniform message routing distributions are considered. This follows from two features of the networks and the routing distributions: network symmetry and similar message routing behavior at all nodes.

LV represents the average number of visits made by a message to *all* communication links. Dividing LV by the number of communication links yields the communication link visit ratios:

$$V_{CL} = \frac{LV}{Numlinks(K, Net-type)} .$$

This quantity can be viewed as a measure of the message intensity supported by a single link. If V_{CL} is near one, then nearly all messages must cross each link at some point along a path to their destination.

Strictly speaking, this simple definition is only accurate if the interconnection network contains only one type of communication link. In a binary tree one would expect the communication traffic on the links at each tree level to be different, leading to different link visit ratios for each level. To accurately analyze interconnection networks with multiple types of communication links, it is necessary to consider the visits to each link type separately.

The visit ratios define the network communication pattern. To fully determine the network workload, the amount of service required by each visit to a device must be specified. Suppose S_i were defined as the average amount of service required by a message during each visit to device i. Then the product $V_i S_i$ would be the *total* amount of service required by an average message at device i. These $V_i S_i$ products are the network workload used in the remainder of our analysis

To simplify this analysis, we further assume that computations require the same mean time S_{PE} at all nodes, and that all links require time S_{CL} to transmit an average message. It should be emphasized that this assumption is *not* required. The succeeding discussion can be applied in its entirety, albeit involving somewhat more arduous symbol manipulation, if each device has a distinct service time.

Solution Techniques

Having established abstract workloads for multimicrocomputer networks, we now consider solution techniques for the resulting models. These techniques range from asymptotic bound analysis [DENN78] to exact solution of product form queueing networks [BASK75], but they all take advantage of network symmetry and the message routing distributions to reduce the model solution time to tractable levels.

Because of our simplifying assumptions and the paucity of prototype multimicrocomputer networks, these techniques are primarily intended to compare interconnection networks and derive order of magnitude performance estimates. More accurate predictions will undoubtedly require more detailed information and extensive simulation studies.

Asymptotic Bound Analysis

Given the workload specified by the $V_i S_i$ products, Denning and Buzen [DENN78] established the following bound on the message completion rate X_0 of a closed queueing network model, assuming only steady state behavior:

$$X_0 \leq \min\left\{ \frac{1}{V_b S_b} , \frac{N}{R_0} \right\} , \tag{5}$$

where

$$V_b S_b = \max_i \left\{ V_i S_i \right\}$$

and

$$R_0 = \sum_i V_i S_i \ .$$

Because of our assumptions about network symmetry and message routing distributions, the number of distinct *VS* products is rather small, normally no more than four or five. This apparent limitation can be turned to advantage; by using the bound (5), equating *VS* products, and solving for the appropriate quantities, it is possible [REED83a] to obtain:

- network sizes where the performance bounds of different networks intersect,
- the ratio of computation quanta needed for two different networks to achieve the same performance,
- the ratio of computation to communication where communication delays limit the message passing rate (i.e., the minimum feasible computation quanta), and
- network performance bounds that are independent of network size.

As illustrations of these last two points, we consider the bidirectional ring once more. The *VS* products for the ring, with uniform message routing, are

$$V_{PE}^{ring} S_{PE}^{ring} = \frac{S_{PE}^{ring}}{K} \quad \text{and} \quad V_{CL}^{ring} S_{CL}^{ring} = \frac{S_{CL}^{ring}(K + 1)}{4K} \qquad (6)$$

with

$$R_0 = S_{PE}^{ring} + \frac{S_{CL}^{ring}(K + 1)}{4} \ .$$

Equating these *VS* products and rearranging terms gives

$$\frac{S_{PE}}{S_{CL}} = \frac{K + 1}{4} .$$

As this critical ratio of computation to communication time, computation delays and communication delays equally limit the message completion rate. If the ratio of computation to communication time for a message falls below this value, communication delays will limit the message passing rate. Furthermore, the ratio depends linearly on K, the number of network nodes. If K were doubled, the ratio of computation time to communication time must also double to prevent communication delays from dominating.

Now consider the limit

$$\lim_{K \to \infty} \left(\frac{1}{\max_i \left\{ V_i S_i \right\}} \right) .$$

This limit defines an absolute upper bound on the message completion rate of a network even if it contained an *infinite* number of nodes. Using the ring with uniform message routing as an example once more, we obtain

$$X_0 < \lim_{K \to \infty} \left(\frac{1}{\max \left\{ \frac{S_{PE}}{K}, \frac{S_{CL}(K + 1)}{4K} \right\}} \right)$$

$$= \lim_{K \to \infty} \frac{4K}{S_{CL}(K + 1)}$$

$$= \frac{4}{S_{CL}} .$$

No ring based system with uniform message routing can pass messages faster than this rate. In general, any interconnection topologies whose message passing rates are bounded above by a constant are unsuitable for large networks.

Product Form Queueing Networks

Heretofore we have obtained only bounds on the network message passing rate. To efficiently evaluate the function $X_0(N)$, the network message passing rate when N messages are present in the network, we require the more restrictive assumptions of product form queueing networks [BASK75].

The most intuitive of the solution algorithms for product form networks, mean value analysis [REIS80], provides a basis for several optimizations and bounding techniques that take advantage of properties specific to interconnection networks. But before proceeding further with its description, we must define some notation unique to queueing network solution algorithms. This notation, summarized in Table II, will be used extensively throughout succeeding sections.

The standard mean value analysis (MVA) algorithm recursively computes $X_0(N)$ from $X_0(N-1)$ and requires $O(NM)$ operations, where M is the total number of nodes and communication links in the network. For networks with $10^4 - 10^6$ devices, evaluation of $X_0(N)$ with a similar number of messages could require over 10^{12} operations, a prohibitive number. In our earlier analysis, we observed that all nodes had the same *VS* product and the communication link *VS* products could be grouped into a small number of types $T - 1$. All devices with the same *VS* product should have the same performance characteristics, so finding these values for one device in each group of distinct *VS* products should suffice. The revised MVA algorithm of Figure II takes advantage of these facts

and requires only $O(NT)$ operations to evaluate $X_0(N)$. Since there are normally no more than four or five unique VS products, an appreciable savings is obtained.

Applying this algorithm to several interconnection networks would show that their message passing rate curves cross in several places. These crossing points are important because they show where it would be advantageous to change from one network to another and the range of network message populations over which one network is preferred. Unfortunately, mean value analysis makes finding these points difficult because it is an *algorithm* and not a formula; the message completion rate for population N cannot be determined without calculating the message completion rates for all populations less than N. Thus, one must enumerate the message completion rates and search for crossing points.

If the requirement for an exact solution to the product form queueing network were relaxed, a formula approximating $X_0(N)$ might be found. The characteristics of such a formula are the subject of the next section.

Balanced Job Bound Analysis

Zahorjan, *et al* [ZAH082] recently established the following upper and lower bounds on the message completion rate of a closed, single class, load independent queueing network:

$$\frac{N}{R_0 + (N-1)V_b S_b} \leq X_0(N) \leq \min\left\{\frac{1}{V_b S_b}, \frac{N}{R_0 + (N-1)V_a S_a}\right\}, \quad (7)$$

where

$$R_0 = \sum_{i=1}^{M} V_i S_i ,$$

$$V_a S_a = \frac{R_0}{M} ,$$

and

$$V_b S_b = \max_{1 \le i \le M} \left\{ V_i S_i \right\} .$$

In the particular case of interconnection networks, this simplifies to

$$R_0 = \sum_{t=1}^{T} O_t \cdot V_t S_t ,$$

$$V_a S_a = \frac{R_0}{\sum_{t=1}^{T} O_t} ,$$

and

$$V_b S_b = \max_{1 \le t \le T} \left\{ V_t S_t \right\} .$$

Figure III illustrates the accuracy of the balanced job bounds for a ring with uniform message routing and unit node and link service times.

The balanced job bounds are attractive for several reasons:

- Only $O(M)$ operations are needed ($O(T)$ for interconnection networks).
- Bounds for a single message population can be obtained independently of any other populations.
- Approximate points of intersection for $X_0(\cdot)$ for different networks can be obtained analytically.

- Appropriate levels of problem parallelism can be estimated.

The last two points are addressed in detail in succeeding sections.

Approximate Intersection Points

Obtaining estimates of the point where network message passing rate curves intersect is easy using the balanced job bounds. Consider two networks with VS products denoted by $V_t S_t$ and $\hat{V}_t \hat{S}_t$. Equating the lower bounds obtained from (7)

$$\frac{N}{R_0 + (N - 1) V_b S_b} = \frac{N}{\hat{R}_0 + (N - 1) \hat{V}_b \hat{S}_b}$$

and solving for N gives

$$N_{low}^{cross} = \frac{\hat{R}_0 - R_0}{V_b S_b - \hat{V}_b \hat{S}_b} + 1 .$$

A similar, though somewhat complicated, approach using the upper bounds gives another approximate point of intersection, N_{high}^{cross}. The true point of intersection, if it exists, lies between these two points.

Consider the relative performance of the ring with uniform routing and a network whose nodes are all connected to a single bus. The VS products for the ring are given by (6); those for the single bus are

$$V_{PE}^{bus} S_{PE}^{bus} = \frac{S_{PE}^{bus}}{K} \quad \text{and} \quad V_{CL}^{bus} S_{CL}^{bus} = S_{CL}^{bus}$$

with

$$R_0^{bus} = S_{PE}^{bus} + S_{CL}^{bus} .$$

Assuming communication delays are the performance limiting factor, equating

the lower bounds on the message passing rates of the ring and the bus gives

$$N_{low}^{cross} = \frac{4K^2 - 9K - 1}{3K - 1}.$$

If there are fewer messages than this in the network, the performance of the single bus is superior to that of the ring.

Optimizing Problem Parallelism

The solution of many important problems can benefit from the judicious application of parallelism, but the key word is judicious. Too little parallelism underutilizes system resources, whereas excessive parallelism results in unnecessary overhead and sometimes even in performance decreases. In principle, it should be possible to determine an optimal level of parallelism for an algorithm on a specified system. In the general case this is extraordinarily difficult, but a simple model of program parallelism can be constructed for networks of the type we have been considering.

Consider a sequential, distributed solution of a problem on an interconnection network. The single circulating message requires service S_{PE} and S_{CL} at the nodes and links respectively. If this message were partitioned into N completely parallel submessages asynchronously cooperating to solve the same problem, optimally, no additional communication or computational overhead would be required, and the mean service times for each submessage would be $\frac{S_{PE}}{N}$ and $\frac{S_{CL}}{N}$. Realistically, there is some overhead involved in combining the results of a parallel computation, and it is unlikely that any problem can be partitioned such that the subproblems are completely disjoint.

Suppose $f(N)$ represents the scaling factor for the mean node service time when the problem solution has been partitioned into N parallel messages. The mean node service time then becomes a function of N:

$$S_{PE}(N) \;=\; f(N) \cdot S_{PE} \;.$$

(This is not a load dependent server but a mean service time that is a function of the total number of customers in the network.) Let $g(N)$ have a similar interpretation for the communication link service times. Then the bottleneck VS product is given by

$$V_b S_b(N) \;=\; \max\left\{ V_{PE} S_{PE}(N)\,,\; V_{CL}^{1} S_{CL}(N)\,,\; \ldots\,,\; V_{CL}^{T-1} S_{CL}(N) \right\}\,,$$

and the function

$$\frac{N}{R_0 \;+\; (N \;-\; 1) V_b S_b(N)} \tag{8}$$

$$= \;\frac{N}{\sum\limits_{nodes} V_{PE} S_{PE}(N) \;+\; \sum\limits_{links} V_{CL} S_{CL}(N) \;+\; (N \;-\; 1) V_b S_b(N)}$$

$$= \;\frac{N}{S_{PE}(N) \;+\; S_{CL}(N) \sum\limits_{links} V_{CL} \;+\; (N \;-\; 1) V_b S_b(N)}$$

represents a lower bound on the rate messages complete when N are in the network. This last qualification is crucial. As the number of messages increases, the amount of useful computation each represents decreases. Because each message represents one Nth of the original problem, (8) must be scaled by $\frac{1}{N}$. This results in

$$\frac{1}{S_{PE}(N) + S_{CL}(N)\sum_{links} V_{CL} + (N-1)V_b S_b(N)} . \quad (9)$$

To find an optimal N, one simply differentiates (9), sets the result equal to zero, and solves for N. At first glance, this appears to be little more than an exercise in symbol manipulation, but precisely this approach has provided some striking insights into an optimal mapping of asynchronous partial differential equations solvers onto multimicrocomputer networks [REED83b].

Additional Modeling Considerations

Increasing the realism of models makes their quantitative evaluation more difficult. Capturing the salient features of a system while eliding irrelevant details is the essence of the modeler's art. Packet switching and simultaneous resource possession are among those features we have elided that deserve comment. More detailed study of multimicrocomputer interconnection networks will undoubtedly require examination of these features.

Packet Switching

For modeling purposes we have assumed that information is passed by messages. In any real system, communication is likely to involve decomposition of messages into fixed length packets. From a practical standpoint, this approach provides several advantages:

- Fixed packet size reduces the amount of buffer space required at nodes intermediate between source and destination.
- Source to destination delay may be reduced by permitting packets to proceed in parallel along multiple shortest paths.

- Communication protocols are simplified.

From a modeling standpoint, the problems of packet switching in a closed queueing network are virtually insurmountable. Delays are involved in message decomposition and reassembly, and the number of packets in the communication network varies over time. No simple method for dealing with these problems is known. Fortunately, all is not as bad as it seems. The communication capacity of an interconnection network is independent of whether the communication paradigm is packet or message based so bounding techniques can be applied in either case. In addition, the estimates for finite message populations provide lower bounds on the communication speed of equivalent networks employing packet switching.

Simultaneous Resource Possession

The simultaneous use of two or more system resources by a single message poses another modeling problem. Suppose the communication controller of each node were capable of transmitting on only one link to which it is connected at any given time. The transmission of a message would require simultaneous use of the source node controller, the receiving node controller, and the communication link connecting them. Clearly, this behavior can adversely affect the maximum message passing rate.

In the worst case, how large could the performance degradation be? If no more than one link connected to each node can be active, scaling the estimates for the case in which all links could be simultaneously transmitting data by the maximum number of connections per node should give a lower bound on the true message passing rate. Whether scaling is needed depends of course on the design of the communication controller and the network communication

protocol. These questions are at a much finer level of detail than we have thus far considered.

Summary

We have described three abstract interconnection network workloads based on different assumptions about internode communication patterns. Using these workloads, we developed computationally efficient exact and approximate solution techniques for queueing network models based on these workloads. With these techniques, it is possible to compare the performance of different interconnection networks processing the same workload, calculate the minimum feasible computation quanta for a network, and determine optimal levels of parallelism.

Acknowledgements

My dissertation advisor, Herbert Schwetman, provided the initial motivation for this work. His insightful comments and insistence on quantifiable results have contributed enormously.

References

BASK75 F. Baskett, K. M. Chandy, R.R. Muntz, and F. G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *Journal of the ACM*, Vol. 22, No. 2, April 1975, pp. 248-260.

BUZE71 J. P. Buzen, "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *Communications of the ACM*, Vol. 16, No. 9, September 1973, p. 527-531.

DENN78 P. J. Denning and J. P. Buzen, "The Operational Analysis of Queueing Network Models," *ACM Computing Surveys*, Vol. 10, No. 3, September 1978, pp. 225-261.

DESP78 A. M. Despain and D. A. Patterson, "X-TREE: A Tree Structured Multiprocessor Computer Architecture," *Proceedings of the Fifth Annual Symposium on Computer Architecture*, ACM Sigarch Newsletter, Vol. 6, No. 7, April 1978, pp. 144-151.

REED83a D. A. Reed and H. D. Schwetman, "Cost-Performance Bounds for Multimicrocomputer Networks," *IEEE Transactions on Computers*, Vol. C-32, January 1983, pp. 83-95.

REED83b D. A. Reed, "Performance Based Design and Analysis of Multimicrocomputer Networks," *PhD Dissertation*, Purdue University, in preparation.

REIS80 M. Reiser, and S. S. Lavenberg, "Mean Value Analysis of Closed Multiclass Queueing Networks," *Journal of the ACM*, Vol. 27, No. 2, April 1980, pp. 313-323.

SMIT82 C. U. Smith and D. D. Loendorf, "Performance Analysis of Software for MIMD Computer," *Proceedings of the 1982 ACM Symposium on Measurement and Modeling of Computer Systems, ACM Performance Evaluation Review*, Vol. 11, No. 4, pp. 151-162. 1980, pp. 328-334.

WITT81 L. D. Wittie, "Communication Structures for Large Multimicrocomputer Systems," *IEEE Transactions on Computers*, Vol. C-30, April 1981, pp. 264-273.

ZAHO82 J. Zahorjan, K. C. Sevcik, D. L. Eager, and B. Galler, "Balanced Job Bound Analysis of Queueing Networks," *Communications of the ACM*, Vol. 25, No. 2, February 1982, pp. 134-141.

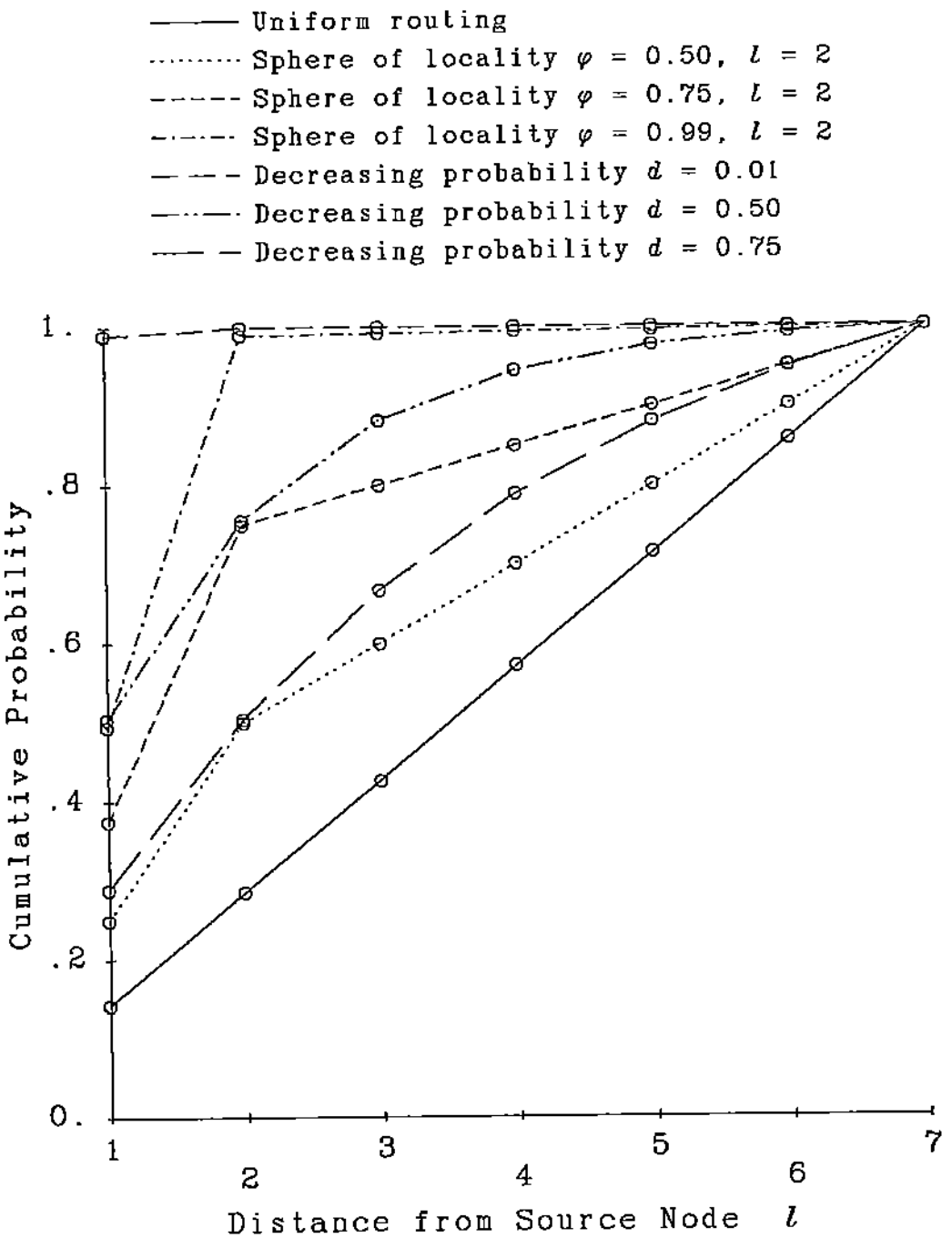

Figure I
Cumulative probability of sending a message to a node l links away from the source node

$\bar{n}_t(0) := 0.0 \qquad t = 1,\ldots,T$

for $N := 1$ to N_{max} do begin

$R_t(N) := S_t\ [\ 1.0 + \bar{n}_t(N-1)\ ] \qquad t = 1,\ldots,T$

$$R_0(N) := \sum_{t=1}^{T} O_t \cdot V_t \cdot R_t(N)$$

$$X_0(N) := \frac{N}{R_0(N)}$$

$X_t(N) := V_t \cdot X_0(N) \qquad t = 1,\ldots,T$

$\bar{n}_t(N) := R_t(N) \cdot X_t(N) \qquad t = 1,\ldots,T$

end

Figure II Modified single class mean value analysis algorithm

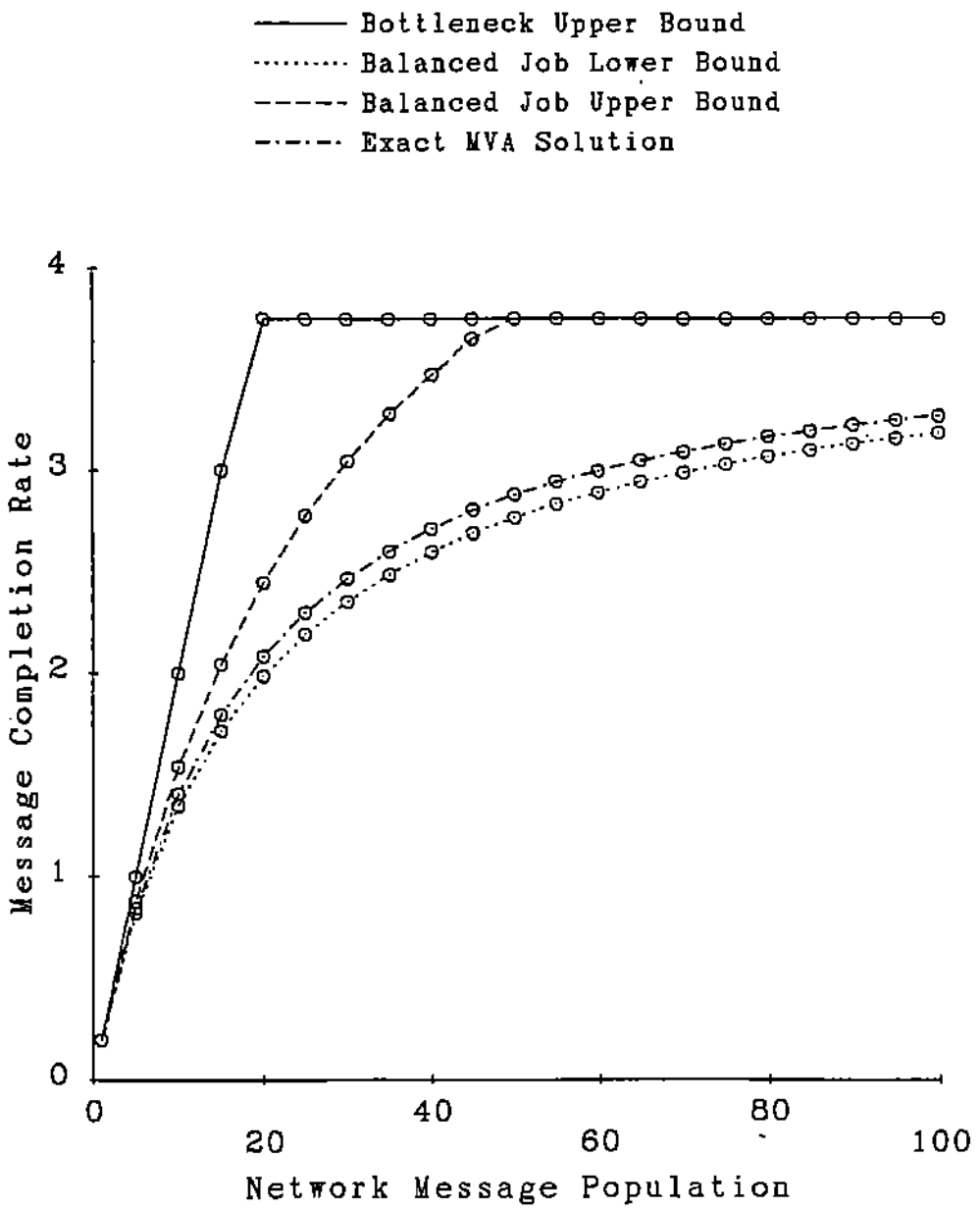

Figure III

15 node bidirectional ring message completion rate
Uniform message routing - unit service times